CentOS 数据库 MySql 安装

版本：mysql-5.7.20-1.el7.x86_64.rpm-bundle

一、 数据库安装

1. 将“mysql-5.7.20-1.el7.x86_64.rpm-bundle.tar”拷到 CentOs 中，并解压 tar 包

   # tar xvf mysql-5.7.20-1.el7.x86_64.rpm-bundle.tar

2. 删除系统中原有的 mysql-libs

   # yum remove mysql-libs

```
[root@www mysql-5.7.20-1.el7.x86_64.rpm-bundle]# yum remove mysql-libs
已加载插件：fastestmirror
正在解决依赖关系
--> 正在检查事务
---> 软件包 mariadb-libs.x86_64.1.5.5.56-2.el7 将被 删除
--> 正在处理依赖关系 libmysqlclient.so.18()(64bit)，它被软件包 2:postfix-2.10.1-6.el7.x86_64 需要
--> 正在处理依赖关系 libmysqlclient.so.18(libmysqlclient_18)(64bit)，它被软件包 2:postfix-2.10.1-6.el7.x86_64 需要
--> 正在检查事务
---> 软件包 postfix.x86_64.2.2.10.1-6.el7 将被 删除
--> 解决依赖关系完成

依赖关系解决

================================================================================
 Package            架构            版本                源                 大小
================================================================================
正在删除:
 mariadb-libs       x86_64          1:5.5.56-2.el7      @anaconda         4.4 M
为依赖而移除:
 postfix            x86_64          2:2.10.1-6.el7      @anaconda          12 M

事务概要
================================================================================
移除  1 软件包 (+1 依赖软件包)

安装大小：17 M
是否继续？[y/N]：y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  正在删除    : 2:postfix-2.10.1-6.el7.x86_64                               1/2
  正在删除    : 1:mariadb-libs-5.5.56-2.el7.x86_64                          2/2
  验证中      : 1:mariadb-libs-5.5.56-2.el7.x86_64                          1/2
  验证中      : 2:postfix-2.10.1-6.el7.x86_64                               2/2

删除:
  mariadb-libs.x86_64 1:5.5.56-2.el7

作为依赖被删除:
  postfix.x86_64 2:2.10.1-6.el7

完毕！
```

3. 安装 common

   # rpm -ivh mysql-community-common-5.7.20-1.el7.x86_64.rpm

4. 安装 libs

   # rpm -ivh mysql-community-libs-5.7.20-1.el7.x86_64.rpm

5. 安装 libs-compat

   # rpm -ivh mysql-community-libs-compat-5.7.20-1.el7.x86_64.rpm

6. 安装 client

   # rpm -ivh mysql-community-client-5.7.20-1.el7.x86_64.rpm

7. 将“libaio-0.3.107-10.el6.x86_64.rpm”拷到 CentOs 中，并安装 libaio

   # rpm -ivh libaio-0.3.107-10.el6.x86_64.rpm

8. 安装 server

   # rpm -ivh mysql-community-server-5.7.20-1.el7.x86_64.rpm

二、 服务相关

1. 启动服务：

   # systemctl start mysqld

2. 查询服务：

   # systemctl status mysqld

```
[root@www mysql-5.7.20-1.el7.x86_64.rpm-bundle]# systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since 二 2018-01-09 15:04:40 CST; 1h 9min ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 2264 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD_OPTS (code=exited, status=0/SUCCESS)
  Process: 2190 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
 Main PID: 2267 (mysqld)
   CGroup: /system.slice/mysqld.service
           └─2267 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid

1月 09 15:04:35 www systemd[1]: Starting MySQL Server...
1月 09 15:04:40 www systemd[1]: Started MySQL Server.
```

   如以上图，说明服务启动成功。

3. 设置开机启动：

systemctl enable mysqld

systemctl daemon-reload

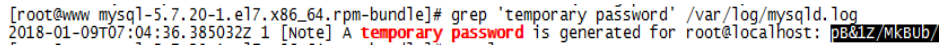
三、 数据库设置

1. 修改 root 密码

（1）查询安装时默认密码：

grep 'temporary password' /var/log/mysqld.log

查询结果如下图：



（2）登录：

mysql -u root -p （1 中的结果“pB&1Z/MkBUb/”）

（3）修改密码：

mysql>set password for 'root'@'localhost'=password('0p;/(OL>');

2. 授予外部访问权限

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY '0p;/(OL>' WITH GRANT OPTION;

3. 防火墙打开默认 3306 端口，参见《防火墙设置》；

4. 修改字符集

vim /etc/my.cnf

添加以下信息：

```
[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option: logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_
ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
```

```
[client]
default-character-set=utf8

[mysql]
default-character-set=utf8

[mysqld]
init_connect='SET collation_connection = utf8_unicode_ci'
init_connect='SET NAMES utf8'
character-set-server=utf8
collation-server=utf8_unicode_ci
#  禁止 MySQL 对外部连接进行 DNS 解析 skip-grant-tables
skip-name-resolve
```

5. 重启服务

   # systemctl restart mysqld

6. 进入查询字符集，查询命名及结果如下：

   mysql> show variables like '%character%';

```
mysql> show variables like '%character%';
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | utf8                       |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | utf8                       |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
```

四、 用户管理

1. 添加用户：

   # mysql -u root –p (输入密码 0p;/(OL>)

   mysql> CREATE USER 'oip2018'@'%' IDENTIFIED BY '9ol.*IK<';

2. 新建数据库：hos_op

   通过 navicat 创建，字符集选 UTF-8

3. 添加权限：

   mysql> grant select,insert,update,delete on hos_op.* to oip2018@"%";

五、 其它

1. 查询数据库端口：

   #show global variables like 'port';

2. 查询密码简略：

   mysql> show variables like '%password%';

```
mysql> show variables like '%password%';
+-----------------------------------------+--------+
| Variable_name                           | Value  |
+-----------------------------------------+--------+
| default_password_lifetime               | 0      |
| disconnect_on_expired_password          | ON     |
| log_builtin_as_identified_by_password   | OFF    |
| mysql_native_password_proxy_users       | OFF    |
| old_passwords                           | 0      |
| report_password                         |        |
| sha256_password_proxy_users             | OFF    |
| validate_password_check_user_name       | OFF    |
| validate_password_dictionary_file       |        |
| validate_password_length                | 8      |
| validate_password_mixed_case_count      | 1      |
| validate_password_number_count          | 1      |
| validate_password_policy                | MEDIUM |
| validate_password_special_char_count    | 1      |
+-----------------------------------------+--------+
14 rows in set (0.02 sec)
```

validate_password_dictionary_file：密码策略文件，策略为 STRONG 才需要

validate_password_length：密码最少长度

validate_password_mixed_case_count：大小写字符长度，至少 1 个

validate_password_number_count ：数字至少 1 个

validate_password_policy：密码策略，默认为 MEDIUM 策略

validate_password_special_char_count：特殊字符至少 1 个

3. 系统内配置及路径：

   配置文件：/etc/my.cnf

   日志文件：/var/log//var/log/mysqld.log

   服务启动脚本：/usr/lib/systemd/system/mysqld.service

   socket 文件：/var/run/mysqld/mysqld.pid

4. my.cnf 加入 skip-name-resolve

   Mysql DNS 反向解析的问题,无论连接数据库的时候采用 IP 直连还是通过域名连接,Mysql 都会尝试 从 IP 解析为 DNS 域名,而目前应用连接数据库是通过 IP 直接连接的,Mysql 仍然会花费较长时间尝试解析 IP 为 DNS 域名。